REMARKS

Reconsideration of this Application is respectfully requested.

Claims 1-30, and 32-34 are pending in the application, with claims 1, 26, 27, 29 and 30 being the independent claims. Claim 31 is canceled without prejudice or disclaimer of the subject matter therein. Claims 30, 32 and 33 are amended. These changes are believed to introduce no new matter, and their entry is respectfully requested.

Based on the above Amendment and the following Remarks, Applicants respectfully request that the Examiner reconsider all outstanding rejections and that they be withdrawn.

Rejections under 35 U.S.C. § 103

In the Action on pages 2-7, section 2, claims 1-29 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,233,356 to Haskell et al. (hereinafter "Haskell"), in view of U.S. Patent No. 6,587,593 to Matsuoka et al. (hereinafter "Matsuoka"). Applicants respectfully traverse the rejection.

As per claim 1, the Action asserts that Haskell teaches all of the limitations of claim 1 except for editing by way of performing an edge operation. In addition to not teaching this limitation of claim 1, Haskell fails to teach several other recited limitations, which the Action neglected to address. Hence, because the Action fails to establish a prima facie case of obviousness, claim 1 is nonobvious and allowable over Haskell.

Claim 1 recites a computer-readable medium having software for editing a decomposed original video sequence, said decomposed original video sequence comprising one or more original camera-motion layers and zero or more original fixed-frame layers decomposed from an original video sequence, said software comprising: code segments for ***editing at least one of said original camera-motion layers*** to obtain modified camera-motion layers such that each frame of ***a composite modified video sequence composed from said modified camera-motion layers*** and said original fixed-frame layers is obtained without editing each frame of said original video sequence, said editing comprising performing an edge operation to one of said original camera-motion layers.

First, Haskell fails to teach **editing** at least one **original camera-motion layer**. Instead, Haskell teaches a general video coding system that codes video objects as scalable video object

layers. Col. 2, lines 35-37. The layers described in Haskell include a video base layer and an enhancement layer. Col. 2, lines 46-48. Haskell does **not** describe or mention **camera-motion layers**. A camera-motion layer, as described by the present specification, is, for example, a video layer that appears to move with the camera as the camera moves. (See, e.g., para. [42] of application). Therefore Haskell does not teach or suggest editing at least one original camera-motion layer.

Second, Haskell fails to teach **editing** such that each frame of **a composite modified video sequence** composed from the modified camera-motion layers and the original fixed-frame layers is obtained without editing each frame of the original video sequence. Instead, Haskell teaches encoding and decoding video objects, but makes **no** mention of **editing** to obtain a composite modified video sequence. In aligning this recitation with Haskell, the Action points to col. 1, line 60 to col. 2, line 11 of Haskell. This alignment is incorrect. The cited passage of Haskell teaches the **coding** of video object planes, but makes **no** mention of editing such that each frame of a composite modified video sequence composed from the modified camera-motion layers and the original fixed-frame layers is obtained without editing each frame of the original video sequence. Therefore, Haskell does not teach or suggest editing such that each frame of a composite modified video sequence composed from the modified camera-motion layers and the original fixed-frame layers is obtained without editing each frame of the original video sequence.

Therefore, because Haskell fails to teach these recited limitations of claim 1, the Action fails to establish a prima facie case of obviousness. Claim 1 is consequently nonobvious and allowable over Haskell.

Further, Matsuoka fails to overcome the deficiencies of Haskell. Matsuoka describes an image processing method that produces illustrations from captured images. Col. 1, line 67 to col. 2, line 1. Matsuoka does **not** teach camera-motion layers. Matsuoka further does **not** teach 1) editing original camera-motion layers; or 2) editing such that each frame of a composite modified video sequence composed from the modified camera-motion layers and the original fixed-frame layers is obtained without editing each frame of the original video sequence.

The motivation provided by the Action for combining Haskell and Matsuoka is insufficient. In particular, the Action cites Haskell at col. 2, lines 12-19, for the motivation to obtain a flexible

editing device. At col. 2, lines 12-19, Haskell states that "one goal of the MPEG-4 standard is to provide a coding scheme that may be used with decoders of various processing power...no known video-object based coding scheme provides such flexibility." The Action asserts that this is a sufficient motivation to combine Haskell in view of Matsuoka.

This motivation, however, is insufficient for the combination as postulated by the Action. In using this motivation, the Action fails to provide any reason why generating an illustration image from a photo or an image using user-specified area edge coordinate positions, as discussed by Matsuoka, would be desirable for video object coding, as taught by Haskell. The Action fails to cite any art for bridging the gap between the divergent teachings of these two references.

Therefore, Haskell and Matsuoka, alone or combination, fail to teach the limitations of claim 1. Applicants respectfully request that the rejection be withdrawn and that claim 1 be allowed.

Claims 2-25 depend from claim 1 and are allowable as being dependent from an allowable claim.

Claims 26 and 27 recite similar limitations as those recited in claim 1, and are allowable for at least the same reasons as those given above with respect to claim 1. Claim 28 depends from claim 27 and is allowable as being dependent from an allowable claim.

As per claim 29, the Action asserts that Haskell teaches all of the limitations of claim 29 except for editing by way of performing an edge operation. In addition to not teaching this limitation of claim 29, Haskell fails to teach several other recited limitations, which the Action neglected to address. Hence, because the Action fails to establish a prima facie case of obviousness, claim 29 is nonobvious and allowable over Haskell.

Claim 29 recites an apparatus for editing an original video sequence, comprising: an object-based video encoder to decompose said original video sequence into a decomposed original video sequence, said ***decomposed original video sequence comprising one or more original camera-motion layers*** and zero or more original fixed-frame layers; a video editor to perform an edge operation to one of said original camera-motion layers and to edit said edge operated original camera-motion layers to obtain a decomposed modified video sequence; and ***an object-based video compositor to compose said decomposed modified video sequence to obtain a composite modified***

video sequence, wherein ***each frame of said composite modified video sequence is obtained without editing each frame of said original video sequence.***

First, as discussed above with respect to claim 1, Haskell does not teach a decomposed original video sequence comprising one or more original **camera-motion layers**.

Second, Haskell fails to teach an object-based video compositor ***to compose said decomposed modified video sequence*** to obtain a composite modified video sequence. Instead, Haskell teaches a video object compositor 900 that composes video from the decoded video object plane (VOP) data for each video object. Col. 4, lines 64-67. Haskell also teaches a scalability post-processor 850 that generates composite video object data from the decoded base layer data and the decoded enhancement layer data. Col. 7, lines 22-27. However, neither video object compositor 900 nor scalability post-processor 850 composes a modified video sequence from a decomposed modified video sequence that comprises an original camera-motion layer. Therefore, Haskell fails to teach an object-based video compositor to compose said decomposed modified video sequence to obtain a composite modified video sequence.

Third, Haskell fails to teach ***each frame of said composite modified video sequence is obtained without editing each frame of said original video sequence.*** Instead, Haskell makes **no** mention of obtaining a composite modified video sequence **without editing each frame** of the original video sequence. Therefore, Haskell does not teach or suggest each frame of said composite modified video sequence is obtained without editing each frame of said original video sequence.

Fourth, as discussed above, Matsuoka fails to overcome the deficiencies of Haskell.

Fifth, as discussed above, the motivation provided by the Action for combining Haskell and Matsuoka is insufficient.

Therefore, Haskell and Matsuoka, alone or combination, fail to teach the limitations of claim 29. Applicants respectfully request that the rejection be withdrawn and that claim 29 be allowed.

Rejections under 35 U.S.C. § 102

In the Action on pages 7-8, section 4, claims 30 and 31 are rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,238,217 to Mirando et al. (hereinafter "Mirando"). Claim

31 is canceled, rendering the rejection moot. Applicants respectfully traverse the rejection of claim 30.

Mirando fails to teach at least one limitation of claim 30. As amended, claim 30 recites a computer-readable medium having software for implementing a video coloring book, wherein said software comprises code segments for ***editing at least one camera-motion layer***. A camera-motion layer, as described by the present specification, is, for example, a video layer that appears to move with the camera as the camera moves. (See, e.g., para. [42] of application).

Mirando fails to teach or suggest editing at least one **camera-motion layer** in a video coloring book. Instead, Mirando teaches animating a picture colored by a user **after** the user is finished coloring. The user selects a picture to color from a limited number of drawings. (Col. 3, lines 7-10.) Once the user has **finished** coloring the picture, the processor then animates the drawing by displaying a sequence of 60-90 existing still frames, using the colors the user has selected. (Col. 3, lines 37-41). There is **no** discussion anywhere in Mirando of a camera-motion layer, or of editing a camera-motion layer to implement a video coloring book. Therefore, Mirando does not teach or suggest editing at least one camera-motion layer in a video coloring book. Consequently, amended claim 30 is allowable.

In the Action on page 8, section 5, claims 32-34 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Mirando, in view of Haskell. Applicants respectfully traverse the rejection.

Claims 32-34 depend from claim 30, and are allowable as being dependent from an allowable claim.

Conclusion

All of the stated grounds of rejection have been properly traversed, accommodated, or rendered moot. Applicants therefore respectfully request that the Examiner reconsider all presently outstanding rejections and that they be withdrawn. Applicants believe that a full and complete reply has been made to the outstanding Office Action and, as such, the present application is in condition for allowance. If the Examiner believes, for any reason, that personal communication will expedite prosecution of this application, the Examiner is hereby invited to telephone the undersigned at the number provided.

Dated:

September 16, 2004

Respectfully submitted,

By _____

Michael A. Sartori, Ph.D.
Registration No.: 41,289
VENABLE LLP
P.O. Box 34385
Washington, DC 20043-9998
(202) 344-4000
(202) 344-8300 (Fax)
Attorney/Agent For Applicant

#572603